Comparing or Configuring Products: Are We Getting the Right Ones?

Nicolas Sannier, Guillaume Bécan, Mathieu Acher, Sana Ben Nasr, Benoit Baudry

Comparing or Configuring Products: Are We Getting the Right Ones?

Nicolas Sannier, Guillaume Bécan, Mathieu Acher, Sana Ben Nasr, and Benoit Baudry
Inria - IRISA Université de Rennes 1
Campus de Beaulieu
35000 Rennes, France
{nsannier, gbecan, mathieu.acher, sana.ben-nasr, bbaudry}@irisa.fr

ABSTRACT

Product comparators and configurators aim to assist customers in choosing a product that meets their expectations. While comparators present similarities and differences between competing products, configurators propose an assisted environment to gradually choose and customize products. The two systems have pros and cons and are inherently different. But both share the same variability information background and operate over a set of (possible) products, typically represented through product comparison matrices (PCMs). A key issue is that current PCMs have no clear semantics, making their analysis and transformations imprecise and hard. In this paper, we sketch a research plan for generating dedicated comparators or configurators from PCMs. The core of our vision is the use of formal variability models to encode PCMs and enables a further exploitation by developers of comparators or configurators. We elaborate on five research questions and describe the expected outputs of the research.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Software configuration management*; D.2.13 [**Software Engineering**]: Reusable Software

General Terms

Theory

Keywords

Product comparators, Product configurators, Product comparison matrices, Variability models, Feature organization

1. INTRODUCTION

Product comparators and configurators have now become common in our daily activities. Whenever one desires to buy a new camera, a computer, a smartphone, a car or a shirt, one can explore, compare and configure a large number of customizable products. In diverse industries, many companies aim at giving customers products closer to their expectations and thus gain higher market shares [31]. Two kinds of systems – product *comparators* and product *configurators* – are usually developed and provided to ease the decision-making process of choosing a product.

On the one hand, product comparators list similarities and differences between two or more than two competing products. These sets of similarities and differences can be presented to customers within *Product Comparison Matrices* (PCMs) (see an example in Figure 1). On the other hand, product configurators offer a controlled and assisted way to choose or customize products [21, 23, 26]. Comparators and configurators target two different use cases, have pros and cons, but both of them share the same *variability* information background and operate over a set of (possible) products.

Variability information contained in PCMs is a rich source of knowledge that both comparators and configurators can exploit to assist customers in choosing the right product. In this paper, we introduce the idea that integrated tools, based on variability modeling and techniques (e.g. [3, 6, 11, 13, 14, 19, 22, 23, 25, 28]), should be developed for generating dedicated comparators or configurators from PCMs.

A key issue is that current PCMs are hard to understand, maintain and exploit. In previous work [29], we analyzed qualitatively and quantitatively 300+ PCMs available on Wikipedia and proposed a set of variability information patterns. This is a first step but several additional research questions have yet to be addressed to fully realize our vision. The empirical understanding of the anatomy and semantics of PCMs, the development of automated techniques to synthesize variability models, and engineering techniques to derive comparators and configurators from variability models are part of the research plan we propose in this paper.

The remainder of the paper is organized as follows. Section 2 introduces a motivating and illustrative example about PCMs, comparators and configurators. We discuss pros and cons (limitations) of the two kinds of systems. We highlight the presence of PCMs and explain our vision. Section 3 presents the anatomy of a commercial PCM and discusses some of the variability patterns that are present in PCMs. Section 4 elaborates a research plan with research questions and expected outcome of the research. Section 5 describes work related to our vision. Section 6 concludes the paper.

2. COMPARING OR CONFIGURING PRODUCTS?

In this section, we present a fictive scenario that illustrates pros and cons of comparators and configurators.

2.1 Searching for their New Car with a Comparator

Alice and Bob live in Montreal (Canada) and want to buy their new car. Bob is a Hockey player, while Alice usually plays keyboards within a rock band. Besides the classic requirements (engine power, color, fuel consumption,

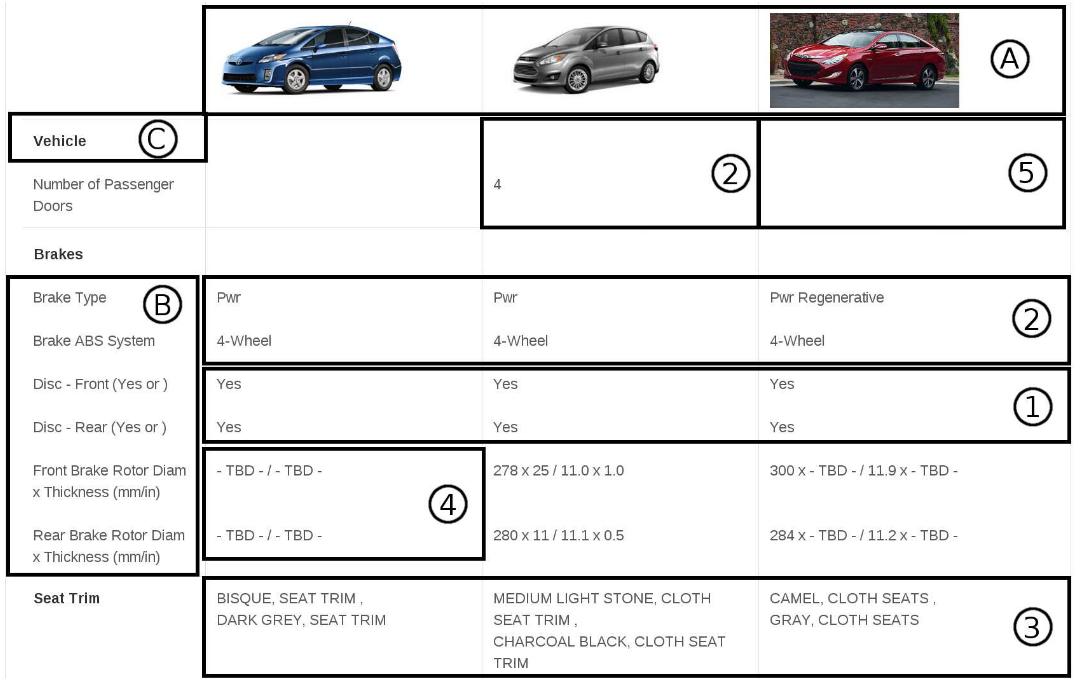

Figure 1: Example of a Product Comparison Matrix in a car comparator

number of doors), they do not have particular requirements relatively to the car brand or its model. However, they do have some "special" requirements that are very important.

- They have a closed garage box. **Rqt1**: The car width must not exceed 1,75m.
- Bob is a relatively large person, as he plays Hockey, and is used to have long trips. **Rqt2**: The car must have a comfortable and adjustable interior.
- When Alice goes to rehearse with her friends, she needs to put down all her stuff in the car. **Rqt3**: The car must have some reasonable cargo capacity.
- Bob enjoys driving using his armrest during his long trips, but Alice does not. **Rqt4**: The car must have independent armrests.
- Alice enjoys listening to music and she spend a lot of time with music on her player. **Rqt5**: The car must have a USB/Dock derivation to plug a media player.

Though these requirements are not major functional requirements for a car, most of them are concrete and may be desirable features and have high customer priority [20].

Many car comparators exist on the Internet. In this section we propose a description of the comparison provided by a Canadian website[1] and illustrated in Fig. 1.

In this comparator, one has to strictly select the year of production, the car brand, the model and its trim (model version). Moreover, the comparator only take into account at most three cars to compare. As a consequence, A user is forced to have a first coarse grain intuition of the vehicle he wants and take a first significant arbitrary decision even before starting the comparison process.

[1]Available online at http://www.autonet.ca/comparenewvehicles#comparenewvehicles-tabs, last access 5th november 2013

With three cars under comparison, the comparator proposes a set of 28 clusters of 104 different features (characteristics). With Different car models, the set of features may vary accordingly to the features presence or absence from the cars. The comparison is presented as a car by feature matrix. These features are organized in two sections, a rather well defined and structured section that contains most of the clusters and a second part containing raw information, which is called "options".

We now observe the comparator's accuracy with respect to Bob and Alice special requirements. We assess the relative position of the requirements evaluation in the matrix:

- **Rqt1**: The answer about the width of the car appears in the "exterior dimensions" cluster, and in particular, at line 96.
- **Rqt2**: Regarding the interior dimensions, they appear in the previous cluster "interior dimensions" and are presented from lines 85 to 92 of the comparison.
- **Rqt3**: For the cargo area dimension, the comparator provides the cargo volumes between lines 103 and 104.
- **Rqt4** and **Rqt5**: For these two requirements, the comparator do not provide any information whereas these two features may represent interesting marketable features.

The comparator does not only provide features information, it also provides a separate factual evaluation of the advantages / disadvantages of the compared cars. However it does not rank these cars, neither it proposes visual ranking of the comparison. It simply proposes a summary of the most obvious differences.

2.2 Configuring their New Car

Bob and Alice still not have completely chosen their new car but they go on configuring several cars from different

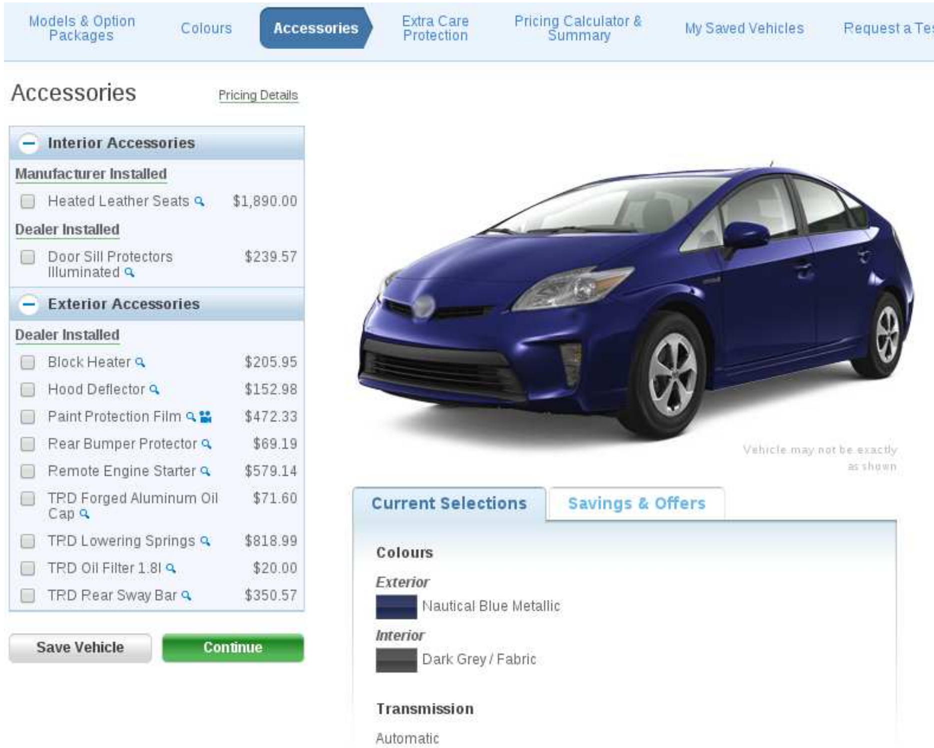

Figure 2: Example of a car configurator

vendors and use car configurators such as the one presented in Fig. 2. These configurators are very similar as they first impose you to chose the model version and package and then tune, within 5 or 6 steps, the remaining options such as color, interior trend, as well as some additional features. In Fig. 2, the scenario is as follows. First, the user chooses one of the car's models and its particular package. The user goes on with the car's color and finishes by choosing interior and exterior items and finally select extra features, which are mostly external items, that are out of the scope of the product specification.

Depending of the vendor's configurator, these steps can be independent or not, but the very large majority of the configuration has already been done while choosing the version of the car and its package. All of them do not propose all of the available features (imposed in the package or not mentioned at all). There exist combinatorial issues that set industrial limits in the product customization, as well as cognitive limits to human variability analysis capability [16]. Though the remaining options still propose a significant variability, the very large majority of the configuration has already been imposed and realized through the initial version packaging.

2.3 On Comparators and Configurators

The fictive scenarios leads us to the following general observations. Comparators and configurators are very simple, convenient and complementary approaches for people who want to have a first glance on different properties over a set of products or about a specific product. They are extensively used within commercial perspectives to provide customers valuable general products information as well as more detailed ones, when available.

Yet user experience is not necessarily satisfactory. Trentin et al. recently discussed five improvement directions for sales configurators that deal with: (1) focused navigation capability, (2) benefit-cost communication, (3) flexible navigation, (4) easy comparison, and (5) user-friendly product space description [31]. The previous scenarios illustrates the limits presented by Trentin et al., not only for configurators but also for comparators. Currently, when comparing products, one must choose between potential candidates, and force a first decision that he may not want to make. User's features are not ordered accordingly to their preferences but presented in a raw. Not all product features are presented neither are subject of analysis. The process of configuring a product is mainly limited to a few number of features and hardwired.

Comparators and configurators actually provide complementary solutions for visualizing and configuring products. Hence both systems can be combined: a comparator is a mean to obtain a first rough intuition of the product he desires ; in a second step, configurator is used to get a more precise product definition and eventually customize it. Comparators can also be integrated as part of a configuration step, when customers focus on some aspect of a product. For example, a comparator of car engines can be activated when configuring the engine.

The current problem faced by practitioners is the lack of comprehensive and systematic solutions for engineering high-quality comparators and configurators that are able to address the previous criteria [21, 31]. A possible unification of the two systems can be realized through the use of *Product Comparison Matrices* (PCMs). In essence, PCMs are a specific form of spreadsheets documenting the features of products under comparison. An example of PCM is in Fig. 1 where three products are compared accordingly to a set of properties (in practice, many more products can be considered and depicted). PCMs are obviously at the heart of product comparators. Configurators also operate over a set of (possible) products and PCMs can be a suitable input data for configurators[2].

Our vision is that integrated tools, based on variability modeling and techniques, should be developed for generating dedicated comparators or configurators from PCMs. Variability models provide a sound foundation for PCMs, comparators and configurators while existing techniques developed in the field can be reused to organize the configuration process, reason about the set of product, etc.

To realize this vision, different research questions have to be addressed. A key issue is to provide and exploit a common, formally defined representation of PCMs through variability models. For this purpose, we need to further understand the anatomy and semantics of PCMs, develop automated techniques to synthesize variability models and exploit the variability models to derive comparators and/or configurators.

3. PRODUCT COMPARISON MATRICES

PCMs abound on the Internet. They can be found in Wikipedia pages but are also visible in Web comparators (see Fig. 1). When comparing two or more products, PCMs are presented to users and list the features supported (or not) by a specific product. Intuitively variability models, like decision or feature models, can formally represent a set of comparable product.

In this section, we illustrate and discuss the variability information patterns (we introduced in [29]) with a PCM found in an existing car comparator.

3.1 Anatomy of a Car Comparator PCM

The car comparator of Fig. 1 compares 3 manually selected cars (A in the figure) against numerous criteria (B in the figure, most criteria are hidden for spacing purpose). This comparison is simply presented as a PCM. We note the presence of different comparison perspectives (C in the figure) materialized by categories in the feature list. We observe the following patterns:

- **Boolean yes/no values** stating the presence or absence of the feature in the product. In the example, Disc front and Disc rear are common features of the 3 products (see 1 in Fig. 1);
- **Single values** whose purpose is to state that the criterion is satisfied and to precise this criterion. In 2 of Fig. 1, the value 4 for the second car states that the car has 4 passenger doors;
- **Multi-values** that enumerate a set of values that satisfy the criterion. The last row of the PCM shows an example (see 3 in Fig. 1). The semantics of multi-values is specific to a given context. A multi-value can have different interpretations and meanings: a mutual exclusion between the values, the availability of all features in the product, etc;
- **Unknown value**: one does not know if the criterion is satisfied. Cells are generally filled with "?", "Unknown" or in the example: TBD (see 4 of Fig. 1);
- **Empty cell**. It is tempting to interpret the empty value as the absence of a feature. Yet it can prevent the product from being selected, despite the uncertainty or the ignorance of the information. 5 of Fig. 1 gives a concrete example: there are obviously doors in a car, but the information is not present.

[2]Configurators are general configuration systems and are not limited to PCMs, i.e., the input of a configurator can be any forms of databases or a logical formula. In particular configurators can handle very large sets of configurations/products that are pointless or hard to enumerate [21,32].

3.2 Commercial PCMs and Variability Information

Interestingly, the variability patterns observed in the car comparator PCM are almost similar to the patterns reported in our previous empirical analysis over Wikipedia PCMs [29]. That is, we also observe Boolean yes/no values, single values, multi-values, unknown values and empty cells. A notable difference is the absence of non homogeneous values for a given criterion (e.g., values can be both Boolean and multi-valued). A possible reason is that PCMs of Wikipedia have been specified by an open and large community while PCMs of comparators are limited to a knowledgeable group of experts, facilitating the use of a common vocabulary. *A systematic analysis of comparators found on the web is needed to confirm these observations.*

Another observation is that variability information in both types of PCMs not only contain boolean values but numerical values, variability inside products and uncertainty. *These types of values challenge state-of-the-art techniques for reverse engineering and reasoning about variability models.*

In this short discussion, we already identify two research directions. In the next section, we elaborate more on the ongoing research challenges that have to be addressed to realize our vision around PCMs, comparators and configurators.

4. RESEARCH PLAN

Our ultimate goal is to create tool-supported techniques for generating dedicated comparators or configurators from PCMs. We now propose research directions related to (1) the understanding, formalization and management of PCMs and (2) the use of variability models as a central formalism for analyzing and configuring a set of product. We also propose a research method and what resulting concrete outputs can be delivered at the end of the research. Fig. 3 summarizes our research plan.

4.1 Research Questions

We aim at addressing the following research questions:

- PCMs are at the heart of the research. It is thus crucial to better understand the syntax, semantics, limits (w.r.t. end-users and maintainers) of PCMs:
  - **RQ1: What is the syntax and semantics of PCMs?** The variability patterns we reported in Section 3 or in [29] have to be refined and formalized and are not claimed to be exhaustive. Other syntactical constructs may be observed in other contexts. Another question concerns the semantics of the patterns, e.g., what does mean an empty value when choosing a product?

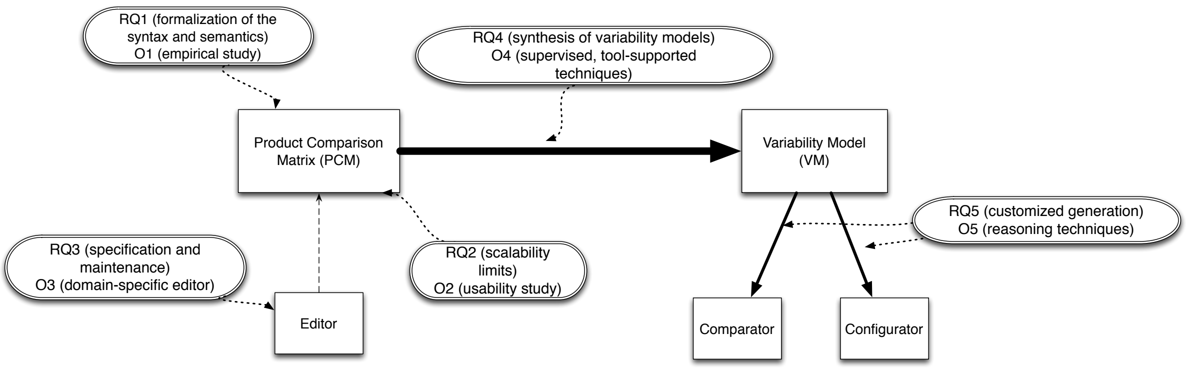

Figure 3: Research questions (RQ) and expected output (O)

  - **RQ2: What are the issues faced by end-users when exploiting a PCM?** End-users *read* PCMs to find the most appropriate product, ideally in a short amount of time. Several factors can limit the user experience: overwhelming users with too much information (i.e., too many products or too many criteria [16]); presenting imprecise information or with unclear semantics, making the PCM hard to understand, etc.
  - **RQ3: How to specify and maintain a PCM?** Maintainers (e.g., contributors of Wikipedia [29], product managers or experts of a domain) *write* PCMs and as such should be differentiated to end-users. Without any canvas, framework, or dedicated tools, practitioners are likely to write ad hoc PCMs with doubtful quality (e.g., with unclear semantics).

- We aim to propose a transition from PCMs to more dependable and assisted comparative (resp. configuration) environments.
  - **RQ4: How to synthesize variability models from PCMs?** Variability patterns found in PCMs have to be translated into a formal variability model using systematic transformation rules. However, the process is likely to be semi-automatic and supervised by practitioners since the variability patterns are subject to interpretation. Scalable synthesis techniques, able to handle numerous products and features, have to be developed and applied in the specific context of PCMs.
  - **RQ5: How to generate configurators or comparators from variability models?** We build upon the vision exposed in [9] that describes a generative process for devising Web configurators. Numerous techniques can be considered for this purpose (see Section 5) but an integrated solution is still missing. We also envision the combined generation of configurators and comparators.

4.2 Research Method and Outputs

To address the five research questions, our research plan and expected outputs are as follows:

- **O1:** As a research method, we plan to empirically study real-world PCMs. Our first work with Wikipedia PCMs is a good starting point but other PCMs can be considered. Such empirical studies can benefit to the understanding of PCMs.
- **O2:** We plan to conduct usability studies to better understand the PCMs limits as well as evaluating pros and cons of providing a configurator and/or a comparator.
- **O3:** As a concrete research output, we plan to develop a domain-specific environment (editor) for specifying PCMs. We hope the resulting PCMs can be easier to visualize, maintain and analyze (e.g., transform).
- **O4:** Supervised, tool-supported techniques to synthesize formal variability models from PCMs are another expected output of the research.
- **O5:** Engineering techniques to generate configurators or comparators from variability models are still to be developed.

5. RELATED WORK

Spreadsheets and PCMs. Considerable research effort has been devoted to the study of spreadsheets [24]. The effort is still ongoing around automated techniques for fault localization and guidelines towards well maintainable spreadsheets [1, 12, 18].

PCMs can be seen as a special form of spreadsheets. However, none of these works consider the specific nature of PCMs and especially the variability information they contain. The empirical study on PCMs of Wikipedia [29] is a first step, but as elaborated in this paper further research effort is needed to understand, formalize and exploit PCMs.

Reverse Engineering Variability Models. The product line research community has shown significant interest in the ability to automatically generate (Boolean) feature models from existing data. Numerous feature model synthesis techniques were proposed [2–4, 17, 30] but they do not consider PCMs as input. Dumitru *et al.* [14], Ryssel *et al.* [27] and She *et al.* [13] process product-by-feature matrices, but only with Boolean values whereas other kinds of values are present in PCMs. Such techniques should be revised to take the variability patterns and kinds of values found in PCMs into account.

In [3], we proposed a semi-automated procedure to support the transition from product descriptions (expressed in a PCM) to feature models. This initial work was conducted on a limited data sample. As a result, we overlooked different aspects of PCMs (e.g. variability patterns, semantics). The research questions exposed in this paper are more general and go beyond variability models.

Variability Modeling. Information in PCMs contains more than simple Boolean values. Products themselves exhibit variability [29]. Boolean variability models fail to formalize such types of values. Extensions of feature models are natural candidates for formalizing PCMs. Attributed feature models [6,11] can handle numerical values and model extra-functional constraints of products. Multi-features [11] can be used to model a part of the variability included in products. Probabilistic feature models [13] can model uncertainty and support soft-constraints.

Product comparators and configurators can benefit from supporting such constraints as users are likely to express preferences instead of definitive choices and propose more flexibility during the comparison and configuration process. Despite these extensions and several studies on variability modeling languages [5,7,8,10,15], the formal connection between variability models and PCMs is still to be defined.

Configuration. Abbasi *et al.* empirically analyzed 111 Web configurators [21] and derived a catalog of good and bad practices. Rabiser *et al.* performed a qualitative study to understand the benefits and drawbacks of guidance capabilities in configuration tools [26]. There are techniques for organizing the configuration process [19], reasoning about decisions [7, 23] and generating user interfaces [25]. Recommender systems [14, 28] or visualization techniques for exploring a set of products [22] can be considered as well when developing configurators or comparators.

6. CONCLUSION

In this paper, we presented our vision towards the use of variability models to devise product comparators or configurators from product comparison matrices (PCMs).

We discussed pros and cons of comparators and configurators through a motivating example applied on Web commercial tools. We showed that PCMs are at the core of comparators and can be inputs of configurators. We described a unified approach based on the synthesis of formal variability models from PCMs. We set five research questions and presented our research methodology related to the understanding, formalization and management of PCMs and the use of variability models for generating comparators and configurators.

In the middle of the bridge between makers and users of product comparators and configurators, perspectives are many. The major challenge is to find the best trade-off between the commercial/industrial combinatorial issues and the product variety paradox [31] that confuses users with too much variability information at a time. We believe that the use of variability modeling techniques and integrated tools to create, maintain, and transform more efficiently PCMs can be leveraged to manage such trade-offs.

7. ACKNOWLEDGMENTS

This work is partially supported by the French BGLE Project CONNEXION.

8. REFERENCES

[1] R. Abraham and M. Erwig. Ucheck: A spreadsheet type checker for end users. *J. Vis. Lang. Comput.*, 18(1):71–95, 2007.

[2] M. Acher, B. Baudry, P. Heymans, A. Cleve, and J.-L. Hainaut. Support for reverse engineering and maintaining feature models. In S. Gnesi, P. Collet, and K. Schmid, editors, *VaMoS*, page 20. ACM, 2013.

[3] M. Acher, A. Cleve, G. Perrouin, P. Heymans, C. Vanbeneden, P. Collet, and P. Lahire. On extracting feature models from product descriptions. In *VaMoS'12*, pages 45–54. ACM, 2012.

[4] N. Andersen, K. Czarnecki, S. She, and A. Wasowski. Efficient synthesis of feature models. In *SPLC'12*, pages 106–115, 2012.

[5] K. Bak, K. Czarnecki, and A. Wasowski. Feature and meta-models in clafer: mixed, specialized, and coupled. In *SLE'10*, LNCS, pages 102–122. Springer, 2011.

[6] D. Benavides, A. Ruiz-Cortés, and P. Trinidad. Automated reasoning on feature models. *CAiSE'05*, 3520:491–503, 2005.

[7] D. Benavides, S. Segura, and A. Ruiz-Cortes. Automated analysis of feature models 20 years later: a literature review. *Information Systems*, 35(6), 2010.

[8] T. Berger, S. She, R. Lotufo, A. Wasowski, and K. Czarnecki. A study of variability models and languages in the systems software domain. *Software Engineering*, 99(PrePrints):1, 2013.

[9] Q. Boucher, G. Perrouin, and P. Heymans. Deriving configuration interfaces from feature models: A vision paper. In *VAMOS'12*, 2012.

[10] A. Classen, Q. Boucher, and P. Heymans. A text-based approach to feature modelling: Syntax and semantics of TVL. *Sci. Comput. Program.*, 76(12):1130–1143, 2011.

[11] M. Cordy, P.-Y. Schobbens, P. Heymans, and A. Legay. Beyond boolean product-line model checking: dealing with feature attributes and multi-features. In *ICSE'2013*, pages 472–481, 2013.

[12] J. Cunha, J. Visser, T. L. Alves, and J. Saraiva. Type-safe evolution of spreadsheets. In *FASE'11*, pages 186–201, 2011.

[13] K. Czarnecki, S. She, and A. Wasowski. Sample spaces and feature models: There and back again. In *SPLC '08*, pages 22–31, 2008.

[14] H. Dumitru, M. Gibiec, N. Hariri, J. Cleland-Huang, B. Mobasher, C. Castro-Herrera, and M. Mirakhorli. On-demand feature recommendations derived from mining public product descriptions. In *ICSE'11*, pages 181–190, New York, NY, USA, 2011. ACM.

[15] H. Eichelberger and K. Schmid. A systematic analysis of textual variability modeling languages. In *SPLC'13*, pages 12–21, 2013.

[16] J. T. Gourville and D. Soman. Overchoice and assortment type: When and why variety backfires. *Marketing Science*, 24(3):382–395, 2005.

[17] E. N. Haslinger, R. E. Lopez-Herrejon, and A. Egyed. On extracting feature models from sets of valid feature combinations. In *FASE'13*, pages 53–67, 2013.

[18] F. Hermans, M. Pinzger, and A. v. Deursen. Detecting and visualizing inter-worksheet smells in spreadsheets. In *ICSE'12*, pages 441–451, 2012.

[19] A. Hubaux, P. Heymans, P.-Y. Schobbens, D. Deridder, and E. K. Abbasi. Supporting multiple

perspectives in feature-based configuration. *Software and Systems Modeling*, pages 1–23, 2011.

[20] N. Kano, N. Seraku, F. Takahashi, and S. Tsuji. Attractive quality and must-be quality. *Journal of the Japanese Society for Quality Control*, 14(2):147–156, 1984.

[21] E. Khalil Abbasi, A. Hubaux, M. Acher, Q. Boucher, and P. Heymans. The anatomy of a sales configurator: An empirical study of 111 cases. In M. Norrie and C. Salinesi, editors, *CAiSE'13*, 2013.

[22] A. Murashkin, M. Antkiewicz, D. Rayside, and K. Czarnecki. Visualization and exploration of optimal variants in product line engineering. *SPLC'13*, 13, 2013.

[23] A. Nöhrer and A. Egyed. C2o configurator: a tool for guided decision-making. *Autom. Softw. Eng.*, 20(2):265–296, 2013.

[24] R. R. Panko. Thinking is bad: Implications of human error research for spreadsheet research and practice. *CoRR*, abs/0801.3114, 2008.

[25] A. Pleuss, B. Hauptmann, D. Dhungana, and G. Botterweck. User interface engineering for software product lines: the dilemma between automation and usability. In *Proc. of EICS'12*, pages 25–34, 2012.

[26] R. Rabiser, P. Grünbacher, and M. Lehofer. A qualitative study on user guidance capabilities in product configuration tools. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, ASE 2012, pages 110–119, New York, NY, USA, 2012. ACM.

[27] U. Ryssel, J. Ploennigs, and K. Kabitzsch. Extraction of feature models from formal contexts. In *FOSD'11*, pages 1–8, 2011.

[28] C. Salinesi, R. Triki, and R. Mazo. Combining configuration and recommendation to define an interactive product line configuration approach. *CoRR*, abs/1206.2520, 2012.

[29] N. Sannier, M. Acher, and B. Baudry. From comparison matrix to variability model: The wikipedia case study. In *ASE*, pages 580–585. IEEE, 2013.

[30] S. She, R. Lotufo, T. Berger, A. Wasowski, and K. Czarnecki. Reverse engineering feature models. In *ICSE'11*, pages 461–470. ACM, 2011.

[31] A. Trentin, E. Perin, and C. Forza. Sales configurator capabilities to avoid the product variety paradox: Construct development and validation. *Computers in Industry*, 64(4):436–447, 2013.

[32] Y. Xiong, A. Hubaux, S. She, and K. Czarnecki. Generating range fixes for software configuration. In M. Glinz, G. C. Murphy, and M. Pezzè, editors, *ICSE*, pages 58–68. IEEE, 2012.